ESCET Version 0.7

User Manual

Thomas R. Schneider

thomas.schneider@ifom-ieo-campus.it

February 28, 2005

Contents

1 Introduction **2**

2 Command Scripts and Data Structure **3**
2.1 Atom sets . . . 3
2.2 Atoms . . . 4
2.3 Residues . . . 4

3 Identification of the rigid part of a macromolecule **6**
3.1 Chorismat Mutase . . . 8
3.2 Asparte Aminotransferase . . . 10
3.3 Tryptophan Synthase . . . 14
3.4 Mersacidin . . . 16
3.5 Ribosomal RNA . . . 20

4 Utilities **28**
4.1 Manipulating PDB-files . . . 28
4.2 SHELXL-related . . . 29
4.3 Analysing models . . . 30
4.4 Comparing things . . . 31

5 Tips and Tricks **33**
5.1 General . . . 33
5.2 ESCET-specific . . . 34

6 Release Notes **36**

Chapter 1

Introduction

ESCET is a script driven program to analyse and compare three-dimensional protein structures. The central idea is to use estimated coordinate uncertainties throughout all calculations. The underlying algorithms are described in a series of papers: [6, 7, 8].

This document contains a short introduction to the program. The easiest way to get started is to look through the different scenarios discussed in chapter 3 and modify one of the examples scripts.

Running the command '`escet -h`' from the UNIX-command line will create an HTML-file called `escet_ref.html` that can be used as a reference manual.

An ESCET-script file can be executed from the UNIX command line using a command of the following form:

```
escet myscript.inp > myscript.log &
```

Diagnostic output will then end up in `myscript.log`, graphical output will be dumped to a postscript file. Scripts to look at the output using RASMOL and or MOLSCRIPT are put into files called `myscript_out.ras` and `myscript_out.mol`, respectively.

Currently, the program is only available as an executable for the Linux operating system.

If you find the program useful, please cite the following paper: Schneider TR: A genetic algorithm for the identification of conformationally invariant regions in protein molecules Acta Cryst D58:195-208 (2002). I would also very much appreciate to receive reprints of papers that report the successful use of ESCET. Thank you.

Please report all problems to me (email: `thomas.schneider@ifom-ieo-campus.it`)

The program and its documentation are copyright Thomas R. Schneider (2000-05).

I would like thank the β-testers (in particular Karl Edman from Uppsala University) for putting up with a semi-functional version of the program and for making lots of useful suggestions. Andrea Cocito (IFOM Milan) has been very helpful in tracking down some very nasty bugs.

Chapter 2

Command Scripts and Data Structure

The program stores data pertaining to atoms in 'atom sets'. An *atom set* can contain an almost unlimited number of atoms, the number of sets is currently limited to 250. Atom sets can be manipulated and analysed using a large number of commands. The commands are normally put in a command script which is then run from a UNIX shell. A typical command script has the form:

```
!
! ESCET-script for making a Bfactor-plot
!

! read information about atoms from pdb-file
! 2lzt.pdb into atom set number one.
aset_read(tset=0,ifile="2lzt.pdb");

! select atoms in residues 5 to 100. Use atom set
! number 0 as the source and atom set number 1 as the target
aset_sele(sset=0,tset=1,sele=(resi in 5-100));

! make a plot of the property 'bfac' of the CA atoms
! vs. residue number for the atoms found in sset 1
rp_plot(sset=1,prop=bfac,aver=ca);
```

Some output will be printed to the screen, graphical output will be dumped to a POSTSCRIPT file.

2.1 Atom sets

Atom sets represent the information that can be found for example in a pdb file. They are basically a collection of atoms with some general information such as name of the protein or the crystallographic unit cell found in the pdb file attached.

Information about atoms can be read from from various file formats. Currently supported formats are: pdb, SHELXL-.ins/.res, SHELXL-.lst. As much information as possible is extracted from the respective files, e.g. if an .lst file contains s.u.'s, these are stored.

Once atoms have been read, sets can be modified by applying selection criteria (command `aset_select`, keyword `sele`).

All commands related to sets of atoms have a syntax of the form `aset_*();` and are described in the reference manual.

2.2 Atoms

Atoms are the 'elementary unit' of the program. An atom has a number of properties that can be used and modified. Examples of such properties are: x, y, z coordinates, atom name, chain identifier, chemical element, B factor etc. pp..

2.3 Residues

In the framework of ESCET, a residue is a set of atoms that share the same chain identifier and the same residue number. To count, a residue must have at least one atom named 'CA'. If properties of entire residues (e.g. the average B factor of the sidechain atoms) have to be stored, the information is kept with the respective C_α-atom.

2.3.1 Secondary structure information

A typical property of a residue is the type of secondary structure element it is located in. This information is stored as a single letter in the *atom property* `ssid` (for secondary structure identifier). of the respective C_α-atom. The single letter code corresponds to the Kabsch&Sander notation as for example used in PROCHECK. The codes and their meanings are given in Table 2.3.1.

code	meaning
'B'	residue in isolated beta-bridge
'E'	extended strand, participates in beta-ladder
'G'	3-helix (3/10 helix)
'H'	4-helix (alpha-helix)
'I'	5-helix (pi-helix)
'S'	bend
'T'	hydrogen-bonded turn
'e'	extension of beta-strand
'g'	extension of 3/10 helix
'h'	extension of alpha-helix
't'	*used in* PROCHECK*, but I don't know what it means*
' '	no secondary structure

Table 2.1: Identifiers used for secondary structure elements

If a pdb-file is read, the program will try to figure out the secondary structure assignment from the HELIX and SHEET records. If you are reading the data from a non-pdb file, a pdb file without the HELIX and SHEET records or if you are not happy with the assignment read, you can modify the *atom property* `ssid` using the `aset_amod` command. The two commands:

```
aset_amod(set=1,ssid=H,sele=(resi in {3-20:34-40}));
aset_amod(set=1,ssid=E,sele=(resi in {24-30:44-50}));
```

will modify the atoms in `set 1` such that residues 3 to 20 and 34 to 40 will become α helical and residues 24 to 30 and 44 to 50 will become β sheet.

Chapter 3

Identification of the rigid part of a macromolecule

Please,
try to roughly understand the paper on the subject: [7] before trying to run the program ...

As it stands now, the analysis of a set of conformers will proceed as follows:

1. Read a bunch of models from one (in the case of NCS-related molecules) or several coordinate files. → `aset_read()`-command.
2. If no standard uncertainties have been read from the coordinate file (as is the case most of the time, unless you are in the fortunate situation of having a SHELXL lst-file from a full-matrix inversion available), some estimates for the coordinate error have to be generated. Normally one would use the B-factor scaled DPI (explained in [6]). → `aset_egen()`-command. The program is pretty clever in extracting the information it needs to calculate the error estimates from the pdb-file. In case, not all the necessary pieces can be found, the information can be added by hand (see section 3.2).
3. Define the conformers that should be compared. In complicated cases, one can also restrict selections to chainids and residue ranges etc. to obtain consistent sets of atoms. (the program tries to figure out the largest consistent set itself, but sometimes needs a bit of help). → `aset_select()`-command.
4. Run the automated error-scaled difference distance matrix analysis. → `ddm()`-command.
5. Look at the results in the html-file or use the automatically generated scripts for LSQKAB,MOLSCRIPT, RASMOL, and PYMOL to create some more intuitive representations.

The following examples are discussed:

Chorismat Mutase (Section 3.1) Compare three NCS-related copies of Chorismat Mutase at 1.3 Å. Make figures that can be used for publication in *Acta Cryst. D*. For more see [7].

Aspartate Aminotransferase (Section 3.2) Compare five different models of Aspartate Aminotransferase. The models have been refined to different resolutions and some of the information necessary to generate standard uncertainties has to added manually. For more see [7].

Tryptophan Synthase (Section 3.3) Two models of Tryptophan Synthase are compared. Both, normal and error-scaled difference distance matrices are displayed for comparison. For more on this analysis, see [6].

Mersacidin (Section 3.4) How to display 36 matrices for two different ways of comparing 6 NCS-related molecules at once. For a change, estimated standard deviations are available from a SHELXL lst-file.

Ribosomal RNA (Section 3.5) Two models of the 16S RNA from the 30S ribosomal subunit are compared. After some necessary massaging of the original pdb-files, difference Distance matrices are displayed and r.m.s.d.-plots are prepared.

3.1 Chorismat Mutase

Comparison of three NCS-related molecules

Here is the script to read the data, divide the atoms into NCS-related copies and run the analysis.

```
!---------------------------------------
! ESCET-script cmut_ddm_5.0.inp
!---------------------------------------

! Read the structure
! ------------------------------------
aset_read(ifile="1DBF.pdb");


! Generate standard uncertainties based on the information
! found in the pdb-file. As the completeness of the data
! was not present in the pdb-file, the value has to be put
! in manually.
! ------------------------------------------------------
aset_egen(sset=0,esd_model=dpiu,cpl=92.0);


! divide the structure into three independent sets of atoms
! --------------------------------------------------------
aset_sele(tset=1,sele=(chainid == A));
aset_sele(tset=2,sele=(chainid == B));
aset_sele(tset=3,sele=(chainid == C));


! calculate an plot the error-scaled difference distance
! matrices for all pairs of molecules
! ------------------------------------------------------
ddm(
  setl={1-3},                ! list of atom sets to look at
  esd_scaled=on,             ! use error-scaling
  lolim=5.0,hilim=10.0,      ! limit for matrix
  xtint = 50,                ! ticks suitable for publication
  ticksfontsize = 14,        ! font suitable for Acta after shrinking
  ssplot=on,                 ! plot secondary structure
  rb_plot=on,                ! plot rigid body description
  pstype=ps,                 ! produce an ps-file on output
  psfname="cmut_ddm_5.0.ps"  ! filename for graphics
);

! finito
stop();
```

To run this script, type:

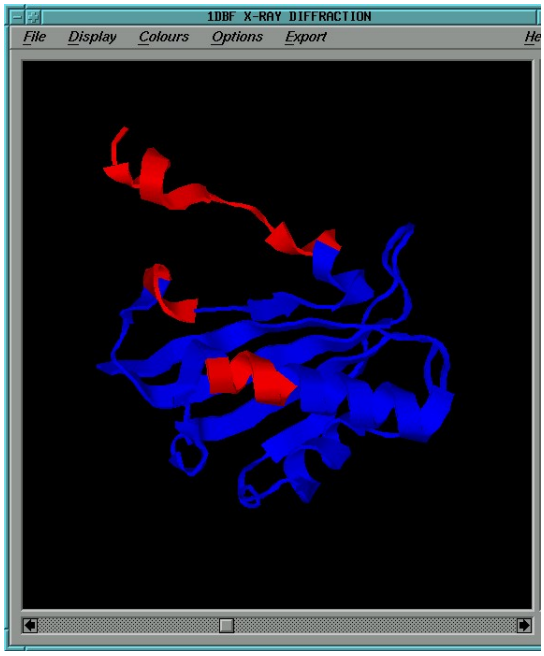
```
> escet cmut_ddm_5.0.inp > cmut_ddm_5.0.log
```

This will dump the difference distance matrices into a file called `cmut_ddm_5.0.ps`. You can look at this file with GHOSTVIEW if your are interested. Otherwise you can check out the log-file `cmut_ddm_5.0_out.html`. Most interesting is probably the graphical representation of the results: files called `cmut_ddm_5.0_out.ras` and `cmut_ddm_5.0_out.mol` have been created that can be used to show the results using RASMOL and MOLSCRIPT respectively.

Typing:

```
% rasmol -script t_ras.ras
```

will bring up a nice little RASMOL -window displaying your molecule with the conformationally invariant atoms in blue and the flexible ones in red (see Figure 3.1.a).

(a) 

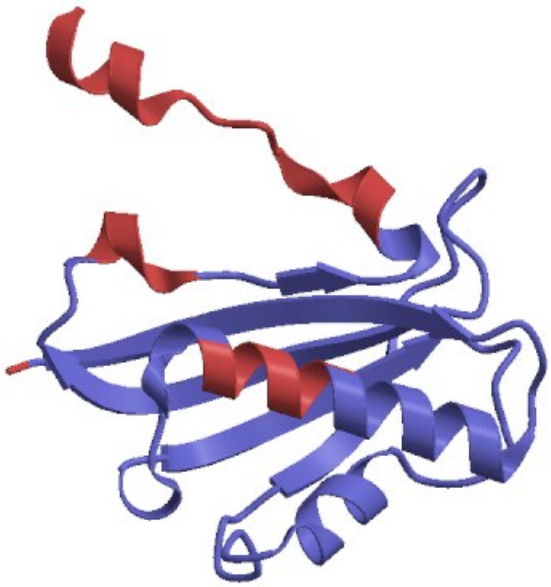
(b) 

Figure 3.1: (a) RASMOL-display of a molecule of Chorismate Mutase with rigid parts mapped onto the molecule in blue, flexible parts in red. (b) The same with MOLSCRIPT.

Typing

```
% molscript -r < t_molscript.mol | render -jpeg > t_molscript.jpg
```

will produce a rendered molscript plot (See Figure 3.1.b). Of course, you may need to put the correct coordinate transformation into the MOLSCRIPT-input file first. If you have the opengl-version of MOLSCRIPT, this is very easy.

With version 0.7, the use of LSQKAB for superposition has become obsolete. The superimposed structures can be found in files called `t_1DBF_[ABC]_mr1.pdb`. A PYMOL-script `t_spos_mr1.pml` is generated that displays the superimposed molecules.

In the first run, `lolim` $= \epsilon_l = 5\sigma$ was used as a lower limit for significant changes. To rerun the whole thing with a different setting of $\epsilon_l = 2\sigma$, all we have to do, is to change the `lolim`-parameter of the `ddm`-keyword and rerun the program.

The bit that is interesting in the log-file, can be found by searching the log-file for the string `'conformationally invariant region'`. Here, the program tells us what it has identified as conformationally invariant:

```
 ==> The following residues form a conformationally invariant region:

     A3-A13, A48-A52, A86-A95.
```

There is some more useful stuff in that part of the file - have a look around.

3.2 Asparte Aminotransferase

Here we read five models from five different files. As the pdb-files are a bit of a mess, we have to put some of the numbers necessary for the calculation of uncertainties by hand:

```
!---------------------------------------
! ESCET-script aatase_ddm1.inp
!---------------------------------------

! Read the models, ignore atoms with zero occupancy
! and generate coordinate uncertainties.
! ------------------------------------------------
aset_read(ifile="7AAT.pdb",sele=(occ > 0.0));
aset_egen(dmin=1.9,npar=27968);
aset_select(tset=1,sele=(chainid == A));

aset_read(ifile="1TAR.pdb",sele=(occ > 0.0));
aset_egen(nobs=36893,npar=26320,cpl=88.5,dmin=2.2);
aset_select(tset=2,sele=(chainid == A));

aset_read(ifile="1AMA.pdb");
aset_egen(dmin=2.3,npar=13892,nobs=17538,cpl=94.4);
aset_select(tset=3,sele=all);

aset_read(ifile="1TAS.pdb",sele=(occ > 0.0));
aset_egen(npar=25408,nobs=17636,cpl=87.9,rfree=30.0,dmin=2.8);
aset_select(tset=4,sele=(chainid == A));

aset_read(ifile="1TAT.pdb",sele=(occ > 0.0));
aset_egen( npar=25408,nobs=18194,cpl=97.0,dmin=3.0,rfree=30.0);
aset_select(tset=5,sele=(chainid == A));

! run difference distance matrix analysis
! ---------------------------------------

ddm(
  setl={1-5},
  esd_scaled=on,
  lolim=2.0,hilim=5.0,
  ssplot=on,
  rbplot=on,
  rb_iterate=3
);

stop();
```

The first run on all the models is used to decide which conformers are identical. For this purpose, a table is printed in the log-file:

	7AAT_A	1TAR_A	1AMA	1TAS_A	1TAT_A
<esd> [A]	0.11	0.32	0.25	0.29	0.39
7AAT_A		**100.0**	73.3	76.2	81.9
1TAR_A			81.9	84.5	86.7
1AMA				**100.0**	**99.9**
1TAS_A					**100.0**
1TAT_A					

This table contains for every pair of models the percentage of elements in the error-scaled difference distance matrix that are smaller than the threshold value of lolim = 2.00. If for a pair

of models, this value is larger than 98.0%, the two models can be considered to be identical. For an explanation see [6].

This table can be used to decide which conformers are redundant and should not be included in the rigid-body analysis in order to not overweight such conformers. If, for a group of conformers, all matrix elements are larger than 98.0with the lowest average esd into the subsequent analysis

Here, for the groups of (7AAT, 1TAR) and (1AMA, 1TAS, 1TAT), the matrix elements indicate that the respective conformers are the same within experimental error.

To find the most precise representative of each group, we can consult another table:

stitle	N_{atom}	Note	N_i	N_{obs} N_{par}	cpl [%]	R_{free} R_{all}	d_{min}	t_{esd}	esd [Angstroem]				
									Mean	Sigma	Min	Max	Mean
7AAT_A	401		6992	60850 27968	96.1	n/a 16.6	1.90	dpiu	**0.107**	0.048	0.030	0.502	|||
1TAR_A	401		6580	36893 26320	88.5	n/a 19.4	2.20	dpiu	**0.326**	0.048	0.222	0.550	|||||||
1AMA	402		3473	17538 13892	94.4	n/a 15.9	2.30	dpiu	**0.252**	0.149	0.041	1.460	|||||
1TAS_A	401		6372	17636	87.9	30.0	2.80	dpiu	**0.295**	0.173	0.041	0.970	|||||
1TAT_A	399		6352	18194	97.0	30.0	3.00	dpiu	**0.392**	0.257	0.051	1.708	|||||||

As 7AAT and 1AMA have the lowest mean coordinate uncertainties, we should continue the analysis with those two conformers as the best representatives of 'their' groups. To run the analysis for those two models only, all we have to do is to change the atoms sets in the `ddm`-command in the above script:

```
ddm(
  setl={1:3},          # <== was setl={1-5}
  esd_scaled=on,
  lolim=2.0,hilim=5.0,
  ssplot=on,
  rbplot=on
);
```

The main result of the analysis is:

Conformationally invariant part - 275/399 atoms = 68.9%
A5-A12, A42-A226, A232-A319.
Flexible part - 124/399 atoms = 31.1%
A13-A41, A227-A231, A320-A410.

Now you can use the corresponding C_α-atoms in your favourite fitting program. Alternatively, ESCET has written a number of PDB-files (named `t_1TAT_A_mr1.pdb` etc. whereby 'mr1' stands for 'moved using rigid region number 1') which contain all conformers superimposed onto the reference conformer and a PYMOL-script that will display the superimposed molecules.

Running

```
% pymol t_spos_mr1.pml
```

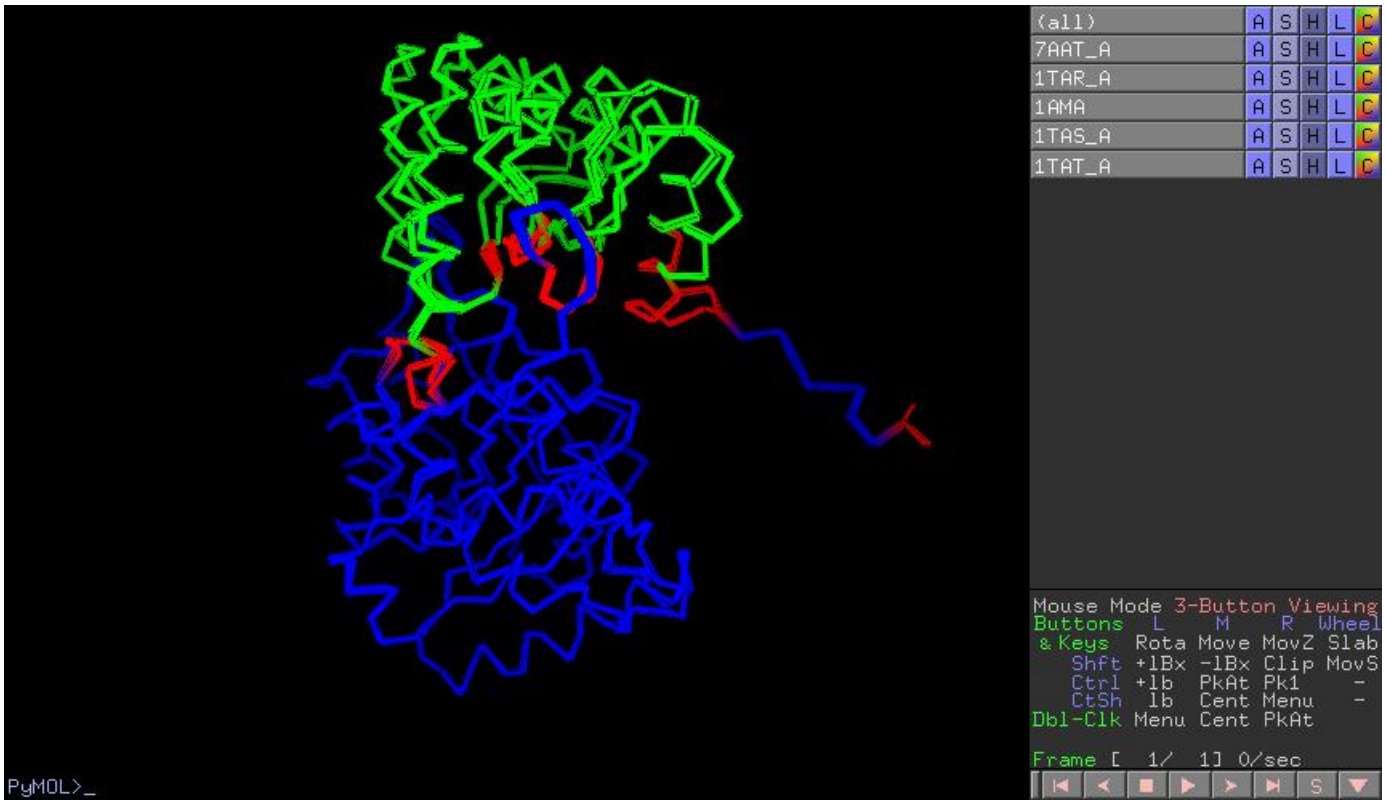


Figure 3.2: PYMOL displaying the result of the analysis of the analysis of the difference distance matrices as a set of superimposed conformers. The first conformationally invariant region is shown in blue (and used for superposition), the second in green. Flexible parts of the molecule are shown in red.

will create the following view shown in Fig. 3.2.

As for the previous examples, there are some premade RASMOL- and MOLSCRIPT-scripts. Running

```
% molscript -r -in aatase_ddm2_out.mol | render -jpeg > aatase_ddm2.jpg
```

will give you a plot like the one in Figure 3.3.

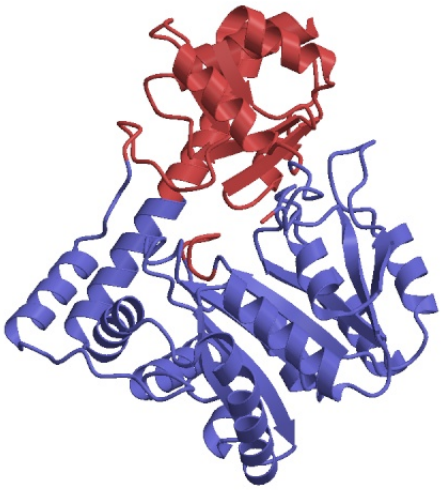


Figure 3.3: MOLSCRIPT-figure of a molecule of Aspartate Aminotransferase with rigid parts mapped onto the molecule in blue, flexible parts in red. For this figure the script `t_molscript.mol` generated by ESCET- the only editing done concerned the orientation matrix.

3.3 Tryptophan Synthase

Tryptophan synthase catalyses the last two reactions in the biosynthesis of tryptophan, the cleavage of indole 3-glycerol phosphate (IGP) to indole and glyceraldehyde 3-phosphate (α-reaction) and the subsequent condensation of indole with serine to form tryptophan (β-reaction) [3]. The reactions take place at two active centers which are separated by a distance of more than 25 Å, but nevertheless are precisely synchronized [1]. In a study aimed at the understanding of the interaction between the two active sites, crystal structures of the enzyme in complex with the substrate analogue 5-fluoroindole propanol phosphate ($TRPS^{F\text{-}IPP}$, pdb-entry 1A50) and in complex with both F-IPP and L-serine ($TRPS^{F\text{-}IPP}_{A\text{-}A}$, where 'A-A' stands for the amino-acrylate that is formed at the β-site under the experimental conditions chosen; pdb-entry 1A5S) were determined [9].

The complete difference distance analysis is described in [6].

```
!---------------------------------------
! ESCET-script aatase_ddm1.inp
!---------------------------------------

! Read the models, ignore atoms with zero occupancy
! and generate coordinate uncertainties.
! ----------------------------------------------
aset_read(ifile="7AAT.pdb",sele=(occ > 0.0));
aset_egen(dmin=1.9,npar=27968);
aset_select(tset=1,sele=(chainid == A));

aset_read(ifile="1TAR.pdb",sele=(occ > 0.0));
aset_egen(nobs=36893,npar=26320,cpl=88.5,dmin=2.2);
aset_select(tset=2,sele=(chainid == A));

aset_read(ifile="1AMA.pdb");
aset_egen(dmin=2.3,npar=13892,nobs=17538,cpl=94.4);
aset_select(tset=3,sele=all);

aset_read(ifile="1TAS.pdb",sele=(occ > 0.0));
aset_egen(npar=25408,nobs=17636,cpl=87.9,rfree=30.0,dmin=2.8);
aset_select(tset=4,sele=(chainid == A));

aset_read(ifile="1TAT.pdb",sele=(occ > 0.0));
aset_egen( npar=25408,nobs=18194,cpl=97.0,dmin=3.0,rfree=30.0);
aset_select(tset=5,sele=(chainid == A));

! run difference distance matrix analysis
! ---------------------------------------

ddm(
  setl={1-5},
  esd_scaled=on,
  lolim=2.0,hilim=5.0,
  ssplot=on,
  rbplot=on,
  rb_iterate=3
);

stop();
```

This script produces the plot shown in Figure 3.4

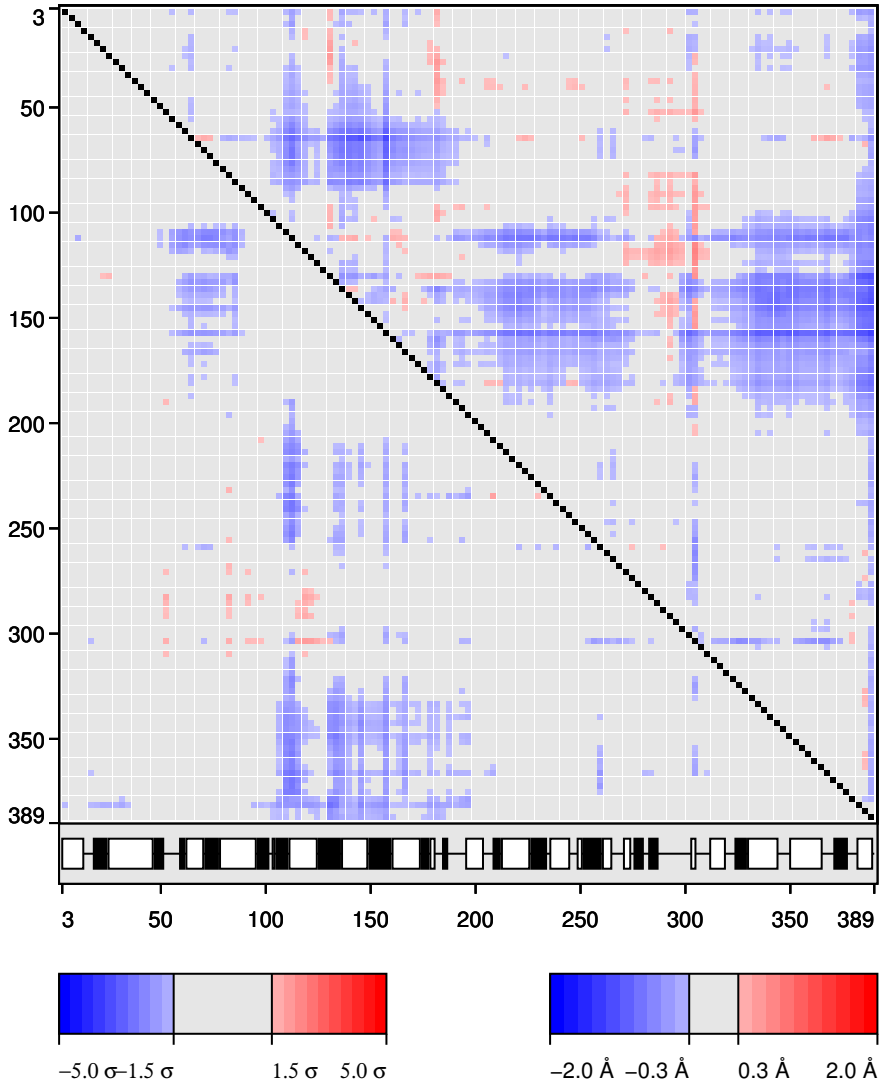


Figure 3.4: Difference distance matrix between the structures of tryptophan synthase in complex with F-IPP ($TRPS^{F\text{-}IPP}$) and in complex with F-IPP and amino-acrylate ($TRPS^{F\text{-}IPP}_{A\text{-}A}$). In the lower left half, the error-scaled difference distance matrix is displayed using upper and lower cutoffs of 1.5 and 5.0σ, respectively. In the upper right half, the ordinary difference matrix is displayed using a lower cutoff of 0.3 Å corresponding to approximately the 1σ level as determined by the σ_A-method. For scaling, an upper cutoff of 2.0 Å has been employed. Both matrices underwent 3×3 binning prior to being displayed. Below the matrix, secondary structure elements are shown schematically: open boxes stand for helices, filled boxes for beta-sheets.

3.4 Mersacidin

Mersacidin is a polypeptide antibiotic containing 20 amino-acids that crystallizes with six molecules in the asymmetric unit. The structure was solved and refined against merohedrally twinned data to 1.06 Å resolution as described in [6]. The conformations of the six molecules are similar to each other with a mean rmsd for all 15 possible pairwise least-squares superpositions (using all $C_\alpha$atoms) of 0.83 Å, as calculated by LSQKAB [5]. Analysis of the superimposed molecules to identify rigid and flexible regions was inconclusive.

For six molecules, there are $((6 \times 6) - 6)/2 = 15$ possible pairwise comparisons. The analysis of the corresponding 15 difference distance matrices allowed the indentification of rigid and flexible parts of the molecule. The analysis is described in [6].

For the six atom sets corresponding to the six molecules, all possible pairwise difference distance matrices can be plotted (on 15 pages of paper) using the following script:

```
! -------------------------------
! ESCET-script: ddm_mers_all.inp
! -------------------------------

! read coordinates and errors from SHELXL list file
! keep only CA-atoms
! ------------------
aset_read(ifile = ./mers_fin_ls.lst,sele=(name == CA));

! prepare six atom sets, one for each molecule
! -------------------------------------------
aset_select(tset=1,sele=(name == CA) and (resi in 101-120));
aset_select(tset=2,sele=(name == CA) and (resi in 201-220));
aset_select(tset=3,sele=(name == CA) and (resi in 301-320));
aset_select(tset=4,sele=(name == CA) and (resi in 401-420));
aset_select(tset=5,sele=(name == CA) and (resi in 501-520));
aset_select(tset=6,sele=(name == CA) and (resi in 601-620));

! automatically produce all error-scaled
! difference distance matrices

ddm(
  setl=1-6,                  ! work on all six sets
  check=loose,               ! loose consistency check
  esd_scaled=on,             ! esd-scaling is used
  lolim=3.0,hilim=5.0,       ! low and high limit for DD-matrix
  psfname=ddm_mers_all.ps    ! postscript output file
);

stop();
```

Following is a script that will produce something similar to Figure 2 in the paper [6], i.e. all difference distance matrices in one plot. The script is somewhat envolved (I hope I will automate this in the future), but I include it here, in case you are facing a similar problem.

```
! --------------------------
! ESCET-script: ddm_mers.inp
! --------------------------

.
.
```

```
Reading coordinates etc. as in ddm_mers_all.inp
.
.


! put first normal difference distance matrix
ddm(
  check=loose,              ! ignore residue number for equivalences
  seta=1,setb=2,
  lolim=0.3,hilim=2.0,
  rb_find=off,
  vx1=100,vy1=560,vx2=170,vy2=630,
  bx1=50,by1=150,bx2=550,by2=750,  ! Set the postcript Bounding Box
  type=both,esd_scaled=off,
  ticks=auto,comments=off,
  xticks=off,xtint=10,
  ticksfontsize=8.0,
  legend=on,
  legendx=100,legendy=200,
  legendw=180,
  pstype=eps,
  psfname=ddm_mers.eps
);

! and the other fourteen, changing the viewport
! position as we go along. We also need to worry
! about putting and not putting ticks.
ddm(seta=1,setb=3,vy1=490,vy2=560,ticks=auto,xticks=off,legend=off);
ddm(seta=1,setb=4,vy1=420,vy2=490);
ddm(seta=1,setb=5,vy1=350,vy2=420);
ddm(seta=1,setb=6,vy1=280,vy2=350,xticks=bottom);

ddm(seta=2,setb=3,vx1=170,vy1=490,vx2=240,vy2=560,xticks=off,yticks=off);
ddm(seta=2,setb=4,vy1=420,vy2=490);
ddm(seta=2,setb=5,vy1=350,vy2=420);
.
.
.

! now the same procedure for the equivalent
! error-scaled matrices
ddm(
  seta=1,setb=2,
  lolim=3.0,hilim=5.0,
  vx1=170,vy1=630,vx2=240,vy2=700,
  type=both,esd_scaled=on,
  limit=150,
  ticks=off,comments=off,
  xticks=off,yticks=off,
  legend=on,
  legendx=340,legendy=200,
  legendw=180
);

ddm(seta=1,setb=3,vx1=240,vx2=310,legend=off);
ddm(seta=1,setb=4,vx1=310,vx2=380);
ddm(seta=1,setb=5,vx1=380,vx2=450);
ddm(seta=1,setb=6,vx1=450,vx2=520);

ddm(seta=2,setb=3,vx1=240,vy1=560,vx2=310,vy2=630);
.
.
```

```
.
ddm(seta=5,setb=6,vx1=450,vy1=350,vx2=520,vy2=420);

ddm(seta=1,setb=1,vx1=100,vy1=630,vx2=170,vy2=700,ticks=auto,yticks=left);
ddm(seta=2,setb=2,vx1=170,vy1=560,vx2=240,vy2=630,yticks=off);
ddm(seta=3,setb=3,vx1=240,vy1=490,vx2=310,vy2=560);
ddm(seta=4,setb=4,vx1=310,vy1=420,vx2=380,vy2=490);
ddm(seta=5,setb=5,vx1=380,vy1=350,vx2=450,vy2=420);
ddm(seta=6,setb=6,vx1=450,vy1=280,vx2=520,vy2=350,xticks=bottom);

! and we are finished. Not so bad.
stop();
```

The figure produced (Figure 3.5) represents a huge amount of information on a single A4 page.

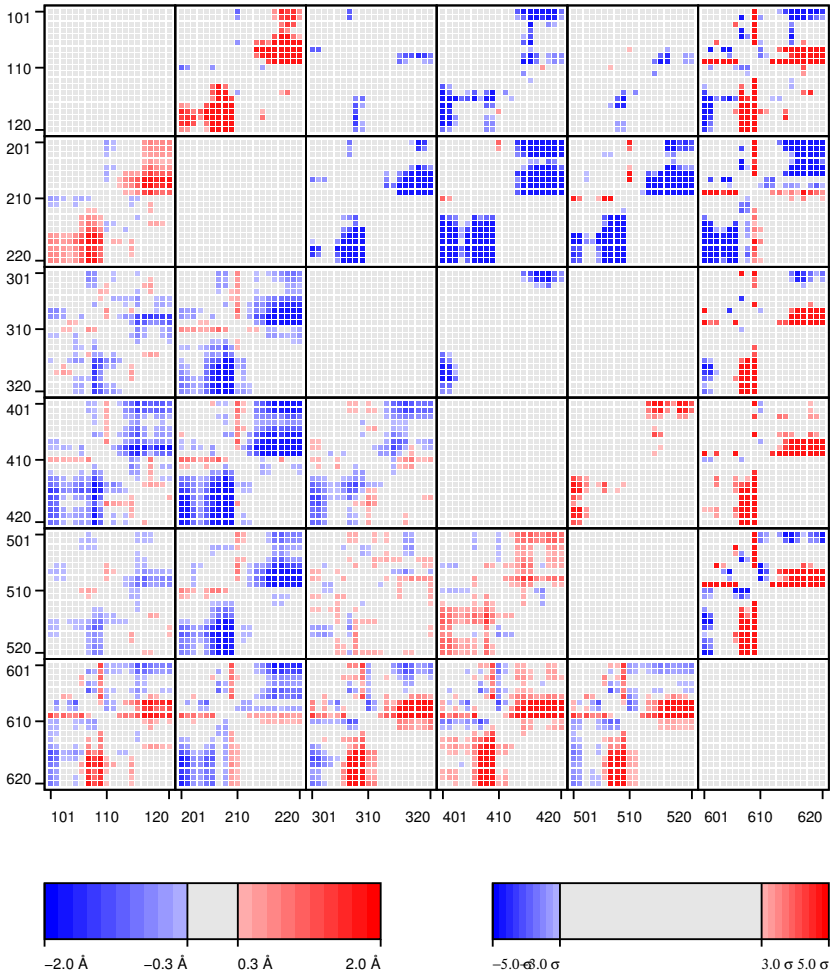


Figure 3.5: Difference distance matrices and error-scaled difference distance matrices for the six molecules of mersacidin. In the lower left triangle, ordinary difference distance matrices for all pairs of NCS-copies are shown. The color-coding is according to the bar on the lower left: all changes in distances smaller than 0.3 Å are shown as gray; differences in distances between 0.3 and 2.0 are shown using a color gradient where red stands for expansion and blue for contraction, light colors represent small changes, dark colors large changes; all differences larger than 2.0 Å are shown as full blue and full red respectively. The blocks in the upper right triangle show the error-scaled difference distance matrices for all pairs of molecules. Here all differences lower than 3.0 times $\sigma(\Delta_{ij}^{ab})$ are mapped to gray. Changes greater than 3.0 and smaller than 5.0 times $\sigma(\Delta_{ij}^{ab})$ are colourcoded using a scheme analogous to the one used for ordinary difference distance matrices.

3.5 Ribosomal RNA

ESCET can not only be used to analyse protein structures but also to compare RNA-molecules using the phophorus atoms as representatives of the nucleotides. Here, the analysis of the RNA compenent of two structures of the 30S ribosomal subunit, one without (1FJF) and one with a drug bound (1IBL) is described.

All ESCET input files mentioned in this section can be found in the `inputs`-directory of the distribution.

3.5.1 Comparing two model

The models with PDB-codes 1FJF and 1IBL contain several stretches of residues, whose residue numbers do not follow the standard numbering scheme of (i.e. consecutive residues have consecutive numbers). Instead there are a number of residue numbers (such as 129 or 190) that are repeated several times - for these residues, in principle the 'Insertion-Id' should be interpreted to obtain sensible results. This special nomenclature causes an enormous amount of complications as most programs do not handle Insertion-Id's properly and the results obtained become a mess. A pragmatic solution to circumvent all these complications is to simply ignore all atoms that have an Insertion-Ids. This is done by ESCET.

The only difference with respect to an analysis of protein structures, is that for the comparison of RNA, the phosporus atoms are used as representatives of the corresponding nucleotides. This fact can be indicated by choosing `rmarker = 'P'` in the aset_read-command when a coordinate file is read (see script below).

The following script will read the coordinate files, assign coordinate error estimates, and find the largest conformationally invariant region of the 16S ribosomal RNA:

```
! ESCET input file 'ribo1.inp'
! ----------------------------
! compare two models of the 30S subunit

! read the first structure and tell ESCET that
! phophorus atoms should be used as the central
! atoms of residues and that only atoms with
! non-zero occupancy should be read
!
aset_read(
  ifile=pdb1fjf.ent,       ! filename
  rmarker="P",             ! central atom of a residue is 'P'
  sele = (occ > 0.0),      ! ignore atoms with occupancy <= 0.0
  notes = "30S",           ! explanatory note about this model
  stitle = "1FJF"          ! short name for the model
);

! generate coordinate uncertainties
!
aset_egen(
  esd_model=dpiu,
  ni=51927,nobs=238205,cpl=94.0,rfree=25.2,dmin=3.05
);

! define the first conformer of the 16S RNA
!
aset_sele(tset=1,
  sele=((chainid == "A") and (part == " "))
```

```
);

! read the second structure
!
aset_read(
  ifile=pdb1ibl.ent,
  rmarker="P",
  sele = (occ > 0),
  notes = "30S + mRNA + Cognate RNA Anticodon Stem loop + Paramomycin",
  stitle = "1IBL"
);

! generate coordinate uncertainties
!
aset_egen(esd_model=dpiu,
  ni=52270,nobs=228284,cpl=89.9,rfree=27.5,dmin=3.11
);

! define the second conformer of the 16S RNA
!
aset_sele(tset=2,
  sele=((chainid == "A") and (part == " "))
);

! determine the conformationally invariant part
!
ddm(
  seta=1,setb=2,  ! compare sets 1 and 2
  ssplot=off,     ! do not plot secondary structure, as this is not defined for RNA
  lolim=1.5,      ! use a lower limit of 1.5 sigma
  dd_plot=on
);

stop();
```

Please note that a lower significance limit of $\epsilon_l = 1.5\sigma$ is used instead of the standard value of 2.0σ. This is done to take into account the fact that the coordinate uncertainties of phosphorus atoms are smaller than for carbon atoms (for which the *dpiu* error model is calibrated).

The results of the run are dumped into an HTML-file that can be displayed with any Internet browser (see Fig. 3.6 and Fig. 3.7). Graphical representations of the results can be displayed with RASMOL and MOLSCRIPT (Fig. 3.8).

stitle	N_{atom}	Note	N_i	N_{obs} / N_{par}	cpl [%]	R_{free} / R_{all}	d_{min}	t_{esd}	esd [Angstroem]			
									Mean	Sigma	Min	Max
1FJF_A	1486	30S	51927	238205	94.0	25.2	3.05	dpiu	**0.357**	0.171	0.144	1.047
1IBL_A	1487	30S + mRNA + Cognate RNA Anticodon Stem loop + Paramomycin	52270	228284	89.9	27.5	3.11	dpiu	**0.386**	0.213	0.130	1.198

Figure 3.6: Table in the HTML-file summarizing the information pertinent to the estimation of coordinate uncertainties.

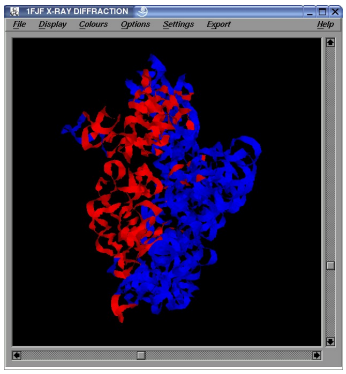
Conformationally invariant part - 1024/1486 atoms = 68.9%

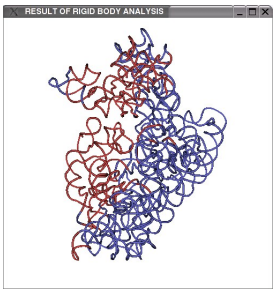
A6-A31, A58-A76, A93-A285, A288-A303, A309-A352, A378-A382, A456-A474, A477-A481, A555-A610, A630-A910, A914-A940, A1025-A1035, A1039-A1043, A1062-A1195, A1237-A1260, A1271-A1298, A1340-A1351, A1368-A1405, A1409-A1492, A1498-A1516, A1519-A1533.

Flexible part - 462/1486 atoms = 31.1%

A32-A57, A77-A92, A286-A287, A304-A308, A353-A377, A383-A455, A475-A476, A482-A554, A611-A629, A911-A913, A941-A1024, A1036-A1038, A1044-A1061, A1196-A1236, A1261-A1270, A1299-A1339, A1352-A1367, A1406-A1408, A1493-A1497, A1517-A1518.

Figure 3.7: Table in the HTML-file showing the results of the search for the largest conformationally invariant region in 16S RNA

(a) 

(b) 

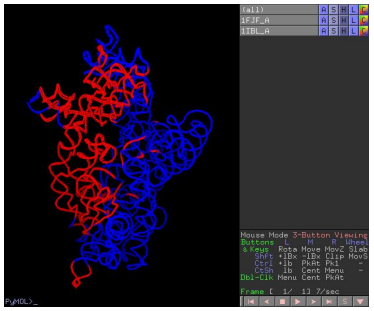
(c) 

Figure 3.8: Graphical display of the conformationally invariant region (in blue) based on the analysis following `ribo1.inp`. (a) in RASMOL using the command `rasmol -script ribo1_out.ras` (b) in MOLSCRIPT using the command `molscript -gl -in ribo1_out.mol` (c) in PYMOL using the command `pymol t_spos_mr1.pml`

3.5.2 Displaying difference distance matrices

Normally, ESCET produces a POSTSCRIPT-file that only contains the error-scaled difference distance matrix. Sometimes, it may be useful to have plot containing both the standard and the error-scaled difference distance matrix. This can be achieved by replacing the `ddm`-command in `ribo1.inp` with the following lines (the complete script can be found in the examples directory as `ribo2.inp`:

```
! ESCET input file 'ribo2.inp'
! ----------------------------
! compare two models of the 30S subunit and
! plot a matrix containing both error-scaled
! and standard difference distance matrix
! elements

.
.
.

! put error-scaled difference distance matrix elements
! to the lower left half of the displayed matrix
!
ddm(
  seta=1,setb=2,
  esd_scaled=on,
  rbfind=on,
```

```
  rbplot=on,ssplot=off,
  lolim=1.5,
  type=lower,
  limit=300,
  vx1=100,vy1=300,vx2=540,vy2=740,
  legend=on,legendx=100,legendy=220,legendw=180,
  ticks=auto, ticksfontsize=10.0, comments=off,
  pstype=eps,
  psfname=ribo2_ddm.eps,
  title="DDM native vs cognate/par"
);

! now put normal matrix into the other corner
!
ddm(
  seta=1,setb=2,
  esd_scaled=off,
  rbfind=off,rbplot=off,ssplot=off,
  lolim=1.0,hilim=5.0,
  type=upper,
  legend=on,legendx=360,legendy=220,
);

! draw a diagonal for clarity
!
ddm(
  seta=1,setb=2,
  type=diag
);

stop();
```

The matrix is shown in Fig. 3.9

3.5.3 Superposition of molecules

During the run of `ribo1.inp`, ESCET generates two scripts that can be used superimpose the two models. `ribo1_out_lsqkab1.csh` is the script that will use LSQKAB to superimpose models using the atoms found to be conformationally invariant in the analysis.

WARNING: The version of LSQKAB coming with CCP4 4.2. can only handle 20000 atoms. Thus the 30S structure can not be handled. There are two solutions: (1) work only with a subset of the coordinate file (such as only P-atoms) or (2) recompile LSQKAB after changing the parameter `NATOM` in the header of `lsqkab.f`.

WARNING: RASMOL will crash when reading large coordinate files written by LSQKAB. The reason for this is that the `CONECT`-records in the PDB-file are messed up. The simplest solution is to delete the `CONECT`-records from the file.

The second command file generated by ESCET `ribo1_out_lsqkab2.csh` will do the superposition based on atoms that are part of the conformationally invariant region *and* which have a coordinate-uncertainty lower than a threshold in the reference model. The default of the threshold is 0.2 Å; the value can be changed using the `RigLoLim-keyword` of the `ddm`-command. To make this script usable, follow the instructions in its header.

Running `ribo1_out_lsqkab1.csh` will superimpose 1048 phosphorus atoms with and r.m.s.d. of 0.67 Å. The moved model is in a file called `pdb1ibl_p_mr.pdb`.

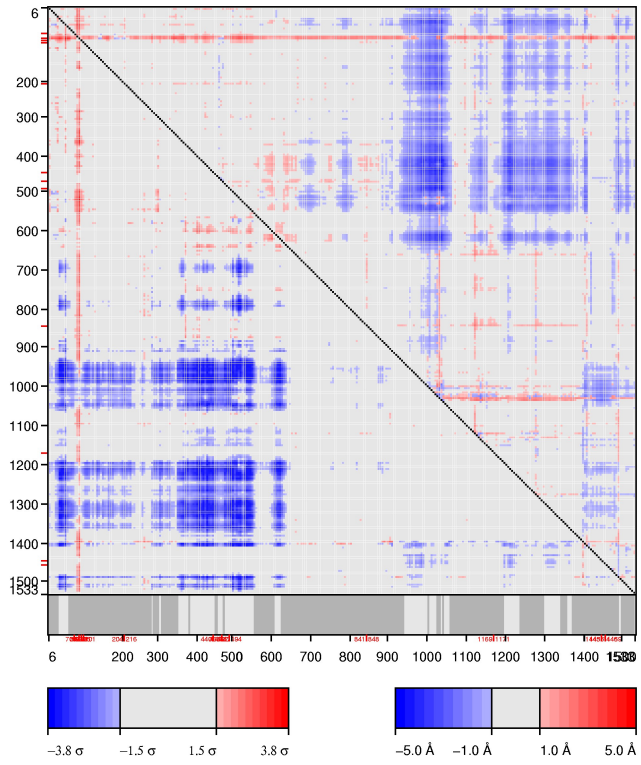


Figure 3.9: Elements of the standard (upper right triangle, right legend) and error-scaled difference distance (lower left triangle, left legend) matrices between two conformers of the 16S RNA shown in one matrix.

3.5.4 Analysis of the superposition

1FJF and the moved version of 1IBL can now be compared in terms of distances between phosphorus atoms:

```
! ESCET input file 'ribo1.inp'
! ----------------------------
! compare two models of the 30S subunit

! read the first structure and tell ESCET that
! phophorus atoms should be used as the central
! atoms of residues and that only atoms with
! non-zero occupancy should be read
!
aset_read(
  ifile=pdb1fjf.ent,       ! filename
  rmarker="P",             ! central atom of a residue is 'P'
  sele = (occ > 0.0),      ! ignore atoms with occupancy <= 0.0
  notes = "30S",           ! explanatory note about this model
  stitle = "1FJF"          ! short name for the model
);

! generate coordinate uncertainties
!
aset_egen(
```

```
  esd_model=dpiu,
  ni=51927,nobs=238205,cpl=94.0,rfree=25.2,dmin=3.05
);

! define the first conformer of the 16S RNA
!
aset_sele(tset=1,
  sele=((chainid == "A") and (part == " "))
);

! read the second structure
!
aset_read(
  ifile=pdb1ibl.ent,
  rmarker="P",
  sele = (occ > 0),
  notes = "30S + mRNA + Cognate RNA Anticodon Stem loop + Paramomycin",
  stitle = "1IBL"
);

! generate coordinate uncertainties
!
aset_egen(esd_model=dpiu,
  ni=52270,nobs=228284,cpl=89.9,rfree=27.5,dmin=3.11
);

! define the second conformer of the 16S RNA
!
aset_sele(tset=2,
  sele=((chainid == "A") and (part == " "))
);

! determine the conformationally invariant part
!
ddm(
  seta=1,setb=2,  ! compare sets 1 and 2
  ssplot=off,     ! do not plot secondary structure, as this is not defined for RNA
  lolim=1.5,      ! use a lower limit of 1.5 sigma
  dd_plot=on
);

stop();
```

The plot (see Fig. 3.10) is stored as a POSTSCRIPT-file under the name `ribo_dist1.ps`. To relate the distance between corresponding atoms to the uncertainties of the coordinates of the atoms involved, a plot containing both the previous plot and a combined estimate of the coordinate uncertainties can be produced:

```
! ESCET input file 'ribo_dist2.inp'
! ---------------------------------
! plot distance between native and the lsq-fitted
! liganded structure vs. residue number and display
! an estimate of the error of that distance.
!

! read two pdb files and declare P-atoms as
! residue markers
!

aset_read(ifile=./t_1FJF_A_mr1.pdb,rmarker=P);
```

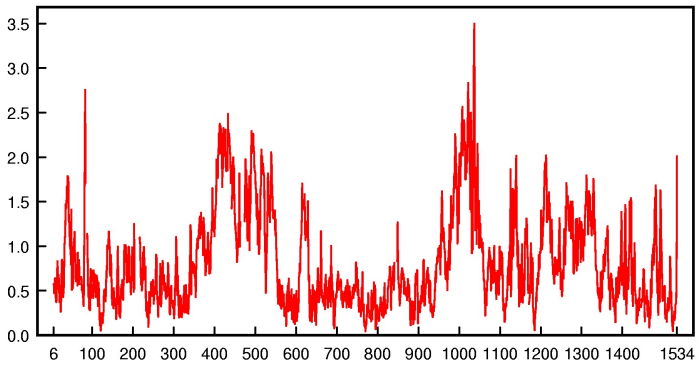


Figure 3.10: Plot of the distance between corresponding atoms in the 16S RNA after superposition based on the conformationally invariant part.

```
aset_egen(
  esd_model=dpiu,
  ni=51927,nobs=238205,cpl=94.0,rfree=25.2,dmin=3.05,
  esd_fac=1.0
);
aset_sele(tset=1,
  sele=((chainid == "A") and (name == "P") and (part == " "))
);


aset_read(ifile=./t_1IBL_A_mr1.pdb,rmarker=P);
aset_egen(esd_model=dpiu,
  ni=52270,nobs=228284,cpl=89.9,rfree=27.5,dmin=3.11,
  esd_fac=1.0
);
aset_sele(tset=2,
  sele=((chainid == "A") and (name == "P") and (part == " "))
);

! plot the distance between corresponding atoms
! as a function of residue number
!
rp_plot(
  rset=1,           ! IFJF is the reference
  sset=2,           ! 1IBL is the working modell
  prop=dist,        ! property to plot
  aver=p,           ! consider only P-atoms
  color=red,        ! color of the curve
  ymin=0,ymax=3.5,  ! min/max for Y-axis
  xtdel1=1500,      ! do no print tick at 1500
                    ! to avoid overlap
  pstype=eps        ! write an eps-file
);

! plot the uncertainty of the interatomic distance
! as a function of residue number
!
rp_plot(
  rset=1,sset=2,    ! as before
  prop=esddist,     ! esd(dist) = sqrt(esd(pos1)^2 + esd(pos2)^2)
  aver=p,           ! consider only P-atoms
  color=blue,       ! color of the curve
  ymin=0,ymax=3.5,  ! min/max for Y-axis
```

```
);

stop();
```

The resulting plot (Fig. 3.11) shows that for a substantial part of the molecule, the differences in atomic positions after superposition based on the conformationally invariant part are on the order of the coordinate uncertainty.

Particularly interesting are the regions around residues 400-500 where the observed differences are significantly larger than the noise and the region around residue 1000 where, despite the large differences between the coordinates, these differences are not necessarily significant as the coordinate uncertainties are rather large (blue curve).

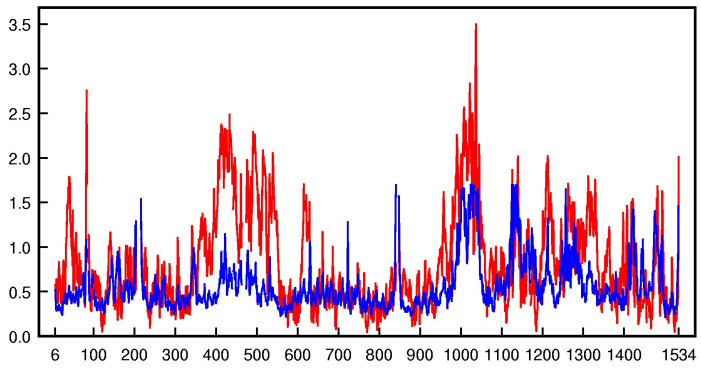


Figure 3.11: Plot of the distance between corresponding atoms in the 16S RNA after superposition based on the conformationally invariant part (red) and the uncertainty of the distance between atoms calculated by error propagation from the coordinate errors (blue).

Chapter 4

Utilities

This chapter contains a loose agglomerate of ESCET-scripts that are useful for all kinds of tasks. All the functions described here are not really supported but may be useful, anyway.

4.1 Manipulating PDB-files

4.1.1 Splitting a pdb-file into separate chain-id's

```
!
! ESCET-script 'epi_split.inp'
!

! read coordinates of CA-atoms from a pdb-file
! -------------------------------------------
aset_read(ifile=./1EQ2.pdb,sele=(name == CA));

! prepare ten atom sets, one for each molecule
! -------------------------------------------
aset_select(tset=1,sele=(chainid == A));
aset_select(tset=2,sele=(chainid == B));
aset_select(tset=3,sele=(chainid == C));
.
.
aset_select(tset=10,sele=(chainid == J));

! now write the all into different files
! -------------------------------------
aset_write(sset=1,ofile=t_A.pdb);
aset_write(sset=2,ofile=t_B.pdb);
aset_write(sset=3,ofile=t_C.pdb);
.
.
aset_write(sset=10,ofile=t_J.pdb);

stop();
```

4.1.2 Change residue names

If you need to change the residue names in a pdb file from 3-letter-code to 1-letter-code, you can use the following commands (in this case for atom set number 2 which happens to be a piece of

RNA):

```
aset_amod(set=2,rtype="G",sele=(resn == "GUA"));
aset_amod(set=2,rtype="U",sele=(resn == "URI"));
aset_amod(set=2,rtype="A",sele=(resn == "ADE"));
aset_amod(set=2,rtype="C",sele=(resn == "CYT"));
```

4.1.3 Modifying a pdb-file with double conformations to use it for MOLSCRIPT

Making pictures of models that contain multiple conformations is not as simple as it could be. MOLSCRIPT does not recognize 'alternate location identifiers' in the pdb-file, so the usual trick is to split the pdb-file into two files containing the different conformations and then to select atoms from the separate models. The following ESCET-script takes a pdb-file applies an offset to all residuenumbers, and then gets rid of all alternate location identifiers. Please note that it is also necessary to sort the atoms as mixed up residue-numbers confuse MOLSCRIPT.

```
aset_read(ifile=ref16r_ls.pdb);             # read the original file
aset_amod(rnoffset=100,sele=(part == B));   # apply offset to all non-B atoms
aset_amod(part=" ",sele=(part <> B));       # remove all alternate location id's
aset_sort(set=0);                           #
aset_write(ofile=ref16r_ls_mod.pdb);
stop();
```

4.2 SHELXL-related

4.2.1 Creating a file to test twinning operators at 3 Å

```
aset_read(ifile=xxx.ins);
aset_write(dmin=3.0,ofile=xxx_mod.ins,otype=res);
stop();
```

The resulting ins-file will still need some editing, but basically implements the following parametrization:

- residues as rigid bodies
- two refinable B-value per residue, one for backbone, one for sidechain
- 3.81 A restraint on CA-CA distance
- SIMU 0.1 0.2 restraint for atoms closer than 4.0 A
- only standard amino acids will be written to output-file
- for technical reasons, atoms will be resorted: N,CA,C,O,CB

4.3 Analysing models

4.3.1 Measuring distances between groups of atoms

The following script first calculates the centers of mass of the different protein chains and then prints the angle between the vectors connecting AB and CD (this angle was used to describe a homotetramer with 222 symmetry):

```
aset_read(ifile=./1hfb.pdb);

aset_vect(
  sel1=(protein and (chainid == A)),
  sel2=(protein and (chainid == B)),
  sel3=(protein and (chainid == C)),
  sel4=(protein and (chainid == D))
);
```

If you select only two groups of atoms, the distance between their centers of mass will be calculated (here the distance between the centers of mass of the imidizale of His282 and the atoms of a PEP-molecule connected by a π-electron system):

```
aset_read(ifile=./1hfb.pdb);

aset_vect(
  sel1=((chainid == A) and (resi == 282) and (sidechain and (name <> CB))),
  sel2=((chainid == A) and (resi == 500) and
         ((name == C1) or (name == C2) or (name == C3) or (name == OC1))),
);
```

4.3.2 Making a Histogram for Ω-angles

Something you always have to put into an atomic resolution paper ...

```
aset_read(
  ifile=./ref16r_ls.lst,
  sele=(protein and (part <> B))
);

rp_plot(
  prop=omega,
  xtrotate=60.0,
  vxp=90,vyp=90,
  pstype=eps
);

stop();
```

4.3.3 Derive some statistics about a model

Another script to fill some tables ...

```
!
! escet-script: stats1.inp
! ------------------------
```

```
! - calculate some statistics on a model
!

aset_read(
  ifile=../feinschliff/ref16s_ls.lst,
  sele=((element <> "H") and (name <> "W"))
);

aset_stat(prop=bfac,
  label="all-full",sele=(occ == 1.0),
  label="all-partial",sele=(occ <> 1.0),
);

aset_select(sele=(element <> "H"));

aset_stat(prop=bfac,
  label="all-full",sele=(occ == 1.0),
  label="all-partial",sele=(occ <> 1.0),
  label="protein-full",sele=(protein and occ == 1.0),
  label="protein-partial",sele=(protein and occ <> 1.0),
  label="water-full",sele=((resn == HOH) and occ == 1.0),
  label="water-partial",sele=((resn == HOH) and occ <> 1.0),
  label="other-full",sele=((not protein and (resn <> HOH)) and occ == 1.0),
  label="other-partial",sele=((not protein and (resn <> HOH)) and occ <> 1.0),
);

aset_stat(prop=esd,
  label="all-full",sele=(occ == 1.0),
  label="all-partial",sele=(occ <> 1.0),
  label="protein-full",sele=(protein and occ == 1.0),
  label="protein-partial",sele=(protein and occ <> 1.0),
  label="water-full",sele=((resn == HOH) and occ == 1.0),
  label="water-partial",sele=((resn == HOH) and occ <> 1.0),
  label="other-full",sele=((not protein and (resn <> HOH)) and occ == 1.0),
  label="other-partial",sele=((not protein and (resn <> HOH)) and occ <> 1.0),
);

! we want to know what 'other' really is, so we write the
! atoms to a file

aset_write(
  ofile="t_others.pdb",
  sele=(not protein and (resn <> HOH))
);

stop();
```

4.4 Comparing things

4.4.1 Compare two models

The `aset_comp`-command will try to do a very simple comparison of two atom sets - **WARNING**: this is very α-test ...

```
aset_read(tset=0,ifile=coord1.pdb);
aset_read(tset=1,ifile=coord2.pdb,sele = (element <> "H"));

aset_comp(seta=0,setb=1);
```

```
stop();
```

4.4.2 Deriving some information from the sequence

Sequences can be read from `pir` or `pdb`-files. For the latter, `SEQRES`-records are used, if available, otherwise the atom list is analysed.

Various numbers like the number of atoms will be printed.

```
seq_read(ifile=../pdb/1HOE.pdb);
```

Chapter 5

Tips and Tricks

5.1 General

5.1.1 Why does LSQKAB not do what I want ?

1. Make sure that the FIT command is always before the MATCH command
2. Make sure that you are not using the same file for REFCRD and WORKCD
3. Make sure that there is no wrongly formatted metal atoms in the pdb-file. A typical error message for this is:

```
fmt: read unexpected character
apparent state: internal I/O
last format: (6X,I5,11X,I4)
lately reading sequential formatted internal IO
Abort
```

5.1.2 How do I use RASMOL ?

There is a very good 'Quick REference Card' at:
http://info.bio.cmu.edu/Courses/BiochemMols/RasFrames/REFCARD.PDF . Another website for RASMOL is at: http://www.umass.edu/microbio/rasmol/ . An overview of a lot of RASMOL related websites can be found at: http://www.rasmol.org .

5.1.3 How do I avoid total confusion when colouring superimposed models in MOLSCRIPT.

A good strategy is to first extract only the pieces of the moved files that you really want:

```
read tmp "pdb1hvy.ent";
copy mol0 require in chain A and in amino-acids;
delete tmp;

read tmp "pdb1hw3_mr.pdb";
copy mol1 require in molecule tmp, in amino-acids and in chain A;
delete tmp;
```

For colouring, first colour everything as flexible and then mark the rigid blocks with another colour:

```
set residuecolour molecule mol*  rgb 1.0 0.35 0.35;  ! flexible

set residuecolour from %61 to %94 rgb 0.5 0.5 1.0;   ! rigid
set residuecolour from %99 to %104 rgb 0.5 0.5 1.0;   ! rigid
set residuecolour from %133 to %147 rgb 0.5 0.5 1.0;   ! rigid
```

The selection for the rigid pieces will only adress existing residues as this is in fact part of the algorithm (an atom has to be existing in all models in order to have a chance to be recognized as rigid).

5.2 ESCET-specific

5.2.1 How do I put a difference distance matrix into a POWERPOINT presentation ?

All difference distance matrices are plotted in POSTSCRIPT -format. This causes Problems when the plots are included into POWERPOINT-presentations (i.e. the plots are not visible on non-POSTSCRIPT devices). I do not know of a program that really converts well between POSTSCRIPT and for example `wmf` (Windows Meta Format).

The solution is to write the matrix that you want in `eps` format and then read it into the program GIMP (`http://www.gimp.org/`). When loading the file you can choose the resolution. I found that 300 dpi is usually good enough. If the resolution is too low you will see strange interference pattern in the matrix. Then save the image in `tif`-format and read it into XV. From XV you can then save the image in `gif`-format (`gif` is very good at storing plots with lots of straight edges in a very compact form). This detour is necessary because GIMP does not support `gif` anymore and XV is not very good at converting POSTSCRIPT to a pixel format.

With newish versions of GIMP and POWERPOINT, PNG is probably the best intermediate format.

5.2.2 Why does the program gets confused by atom names used for my co-factor ?

Sometimes atoms of co-factors have rather strange names. One examples is NADP. The pdb says about this at the following location:

http://www.rcsb.org/pdb/docs/format/pdbguide2.2/guide2.2_frame.html

in large het groups it sometimes is not possible to follow the convention of having the first two characters be the chemical symbol and still use atom names that are meaningful to users. A example is nicotinamide adenine dinucleotide, atom names begin with an A or N, depending on which portion of the molecule they appear in, e.g., AC6 or NC6, AN1 or NN1.'

5.2.3 After I start ESCET, my computer starts swapping. Why ?

As it stands, the program is not really optimized in terms of memory consumption. A matrix for 2000 x 2000 atoms will take 2000 x 2000 x 4 Bytes (float) = 16 MBytes of memory. Comparing 6 such models will give 16 matrices, i.e. roughly 256 MBytes of memory are necessary to run such a problem. But, hey, memory is cheap ...

5.2.4 How can I run ESCET, on homologous (i.e. non-identical molecules) models ?

Extending ESCET to work on homologous structures is one of the things I plan to include in the future. Right now, however, you would need to allign the molecules externally and then compare the matching segments. Let's say, you have molecules A and B and you have matches like:

```
A10-A20 <-> B15-B25
A25-A30 <-> B42-B47
A50-A70 <-> B48-B68
```

Then you could do selections like:

```
aset_sele(tset=1,sele=(resi in {10-20:25-30:50-70}));
aset_sele(tset=2,sele=(resi in {15-25:42-47:48-68}));
```

and run a normal `ddm`-analysis using the keyword `check = loose` to allow non-identical residue names for atoms being compared.

```
ddm(check=loose,seta=1,setb=2, .... );
```

Please note that with `check=loose`, it is not possible to do an interative analysis of your ensemble to find domains. In fact, to make the script run, you also have to specify `rb_iterate=0`:

```
ddm(check=loose,rb_iterate=0,seta=1,setb=2, .... );
```

5.2.5 How much memory will my ESCET-job require ?

The necessary memory in units of Bytes can be calculated via:

$$(natom * natom) * ((nmodel * nmodel)/2 - nmodel) * 8Bytes,$$

where *natom* is the number of atoms and *nmodel* the number of models

Chapter 6

Release Notes

Known Bugs

- For multiple chain analyses, the rigid body display on the bottom of the DD-matrices is sometimes messed up. However, the calculations and the RASMOL, and MOLSCRIPT files are correct.

Version 0.7 28-Feb-2005

Version 0.7 introduces a number of major improvements:

- The Kabsch-Algorithm for least-squares superposition [4] has been implemented. The program will now automatically superimpose the conformers analysed by applying the Kabsch-algorithm to the different conformationally invariant regions. The tranformed coordinates are written to pdb-files for later use. The pdb-files are named t_XXXX_C_spos_mr#.pdb, where XXXX is the pdb-code of a conformers, C its chain name. '#' is a number indicating the conformationally invariant region used for superposition.

- The genetic algorithm for the identification of conformationally invariant regions can now be run in an iterative fashion. This allows to not only find the largest conformationally invariant region, but also other rigid parts of a molecule. For example, src-kinase can be divided into its four domains using this approach (see [8] for a explanation). The default number of iterations is 3 and can be changed using the `rb_iterate`-keyword in the `ddm`-command. Setting `rb_iterate=0` should go back to the pre version 0.7 behaviour of the `ddm`-command.

  Please note that for various technical reasons, the iterative analysis can only be performed on different conformers of the same molecule.

- To be able to write complete models after analysis and superposition for visualization purposes and for further analysis, a slight change in the general approach used in ESCET had to be introduced. Previously, the atoms (e.g. CA's) had to be selected before entering the `ddm`-command. Now this selection is done within the `ddm`-command using the `sele`-keyword with a standard selection string. The default for this selection is to select CA-atoms. Thus for most users, nothing will change except that they have to remove the explicit selection of CA-atoms in earlier parts of ESCET-scripts.

Users interested in phophorus atoms as representatives of RNA-structure, can use `sele = (name == P)` to switch to phosphorus atoms.

People interested in a analysing complex mixtures of atoms can go back to the old behaviour by stating `sele = all`.

- Scripts for visualization of results using pymol are now generated. t_spos_mr1.pml will show all conformers superimposed using rigid region number 1, t_spos_mr2.pml will shown the super position on rigid region number 2 etc.
- Various pieces have been added to the html-file, e.g. a table containing statististics about coordinate uncertainties of all conformers, a table containing rmsd-values for all least-squares superpositions done etc. There is also a hyper-linked table of contents at the top of the file.
- When atoms with zero-occupancy are read, a warning is generated.

Version β 0.6e, 14-Jun-2004

- Section on the comparison of ribosomal RNA was added to the manual.

Version β 0.6d, 08-Jun-2004

- Notes can now be attached to models read from pdb-files. Use the `notes`-keyword of the `aset_read`-command to make notes.
- An overview of all conformers being analysed is now displayed as a table in the HTML output file.

Version β 0.6, 26-Feb-2004

- many small fixes.

Version β 0.5, 13-Oct-2003

- ESCET can now be started by giving the name of the `.inp` as the first argument. This will generate output files using filenames related to the input-file name.
- In addition to the loggin information dumped to the Terminal, the program now generates an output file in HTML-format that can be printed, cut-and-pasted etc.
- Several new entries have been added to the 'Utilities'-section of the manual.

Version β 0.3, 24-Jan-2003

- Loads of small fixes.

Version β 0.2f, 27-Jun-2003

- The Bounding Box for encapsulated postscript can now be set explicitly using `bx1,by1,bx2,bx1` keywords of graphics-related commands.
- Keywords `xtdel1` and `xtdel2` can be used to specify up to two tickmarks that will be explicitly deleted if, e.g. if tickmarks overlap on the lower or upper end of a scale.
- Several problems with tickmarks for multi-chain plots have been fixed.
- `title` and `frametitle` are now separate concepts. `frametitle` is the title that will be put into the frame of a plot, `title` is the title that will be put on top of the actual figure. Use keywords `title` and `frametitle` to set the values. If several plots are overlayed in one figure, the titles will be overlayed as well. Setting `title=" "` should help in such cases. For `rp_plot`, the defaults are: `title=""`, `frametitle="something invented by the program"`

Version β 0.2e, 17-Apr-2002

- Two keywords `patchlen` and `minfraglen` are now available to adjust the polishing of a solution of the rigid body search. See `escet_ref.html` for details.
- Diagnostics for automatic consistency checks has been improved

Version β 0.2d, 11-Apr-2002

- Some problems with automatic consistency checking of atom lists were fixed.
- The Pairwise comparison table now contains a line with the mean estimated error for the atoms that are actually compared. This allows to easily pick out the best determined atom set from a set of redundant conformers.
- STDOUT is now flushed regularly to make monitoring easier.

Version β 0.2c, 1-Mar-2002

- A warning is now given, if a pdb-file contains B-values that are smaller or equal 0.0 - such B-values would mess up the error-estimation.
- The manual is again available in pdf-format !
- The plotting if DD-matrices for multi-chain problems is still not perfect (loads of book-keeping to do ...) but at least does not hang anymore.
- The program will now give an estimate on the memory it will use. If this estimate is close to the physical RAM of your computer, try to use a mashine with more memory to avoid swapping.

Version β 0.2, 1-Feb-2002

- Rewrote documentation to reflect zillions of changes.
- Major improvements on RASMOL-,MOLSCRIPT-, and LSQKAB-scripts generation.
- Substantial cleanup on the log-file.

Version β 0.1g, 5-Jun-2001

- Fixed some problems with MOLSCRIPT-output of the rigid body finder.
- Argument `dd_plot=off` in keyword `ddm` will switch off the postscript output.

Version β 0.1f, 25-May-2001

- The conformationally invariant part of a molecule can now be automatically determined using a genetic algorithm. Right now, no parameters are accessible from the interface (changing parameters still has to be done in the source ...). But using `rb_find = on` in the `ddm`-command, will run the algorithm using a reasonable set of default parameters. Setting `rb_plot = on` and `ss_plot = on` will put the information also into the plotted difference distance matrices (rigid parts are marked as dark gray, flexible as light gray).
  The program also dumps a bunch of files to use as input to MOLSCRIPT,RASMOL, and LSQKAB. I will work hard on documenting these, promised. `molscript -gl < t_molscript.mol` will give a pretty picture in many cases (blue for rigid, red for flexible). If it doesn't work, it will at least be a good template ...
  If more than two atom sets were selected using the `setl`-keyword in the `ddm`-command, the automatic interpretation will take all models into account automatically !
- In previous version the Cruickshank error estimate was calculated deriving some of the input numbers from the current atom set. If the calculation was done after removing some atoms or only retaining the CA's, this could give wrong results. The program now remembers the numbers from the original pdb-file and uses those as default values.
- `aset_egen` now has a keyword `esd_blim` to select the lowest accepts B-value.
- if secondary structure is plotted, now the data for this come from a reference set. The number of this set defaults to the first set selected, but can be changed using the `setr` keyword of `ddm();`
- `sele` now works properly with `chainid == " "`.
- `aset_write` can now write selected sets of atoms.
- The program can now hold 100 atom sets.
- Some cleanup on eps output files.
- Started a chapter on tips & tricks.

Version β 0.1e, 9-Jan-2001

- Chorismate Mutase was included as an example into the user's manual.
- Hopefully, the use of Equation 26 or 27 from Cruickshank's paper [2] is now somewhat clearer
- When reading coordinates from a file using the `aset_read`-command, atoms can now be selected directly, i.e. if you only want CA-atoms from a file called `fname.pdb` use: `aset_read(ifile="fname.pdb",sele=(name == "CA"))`.

Version β 0.1d, 14-Nov-2000

- A number of things concerning the refinement are now extracted from the pdb-file and passed on to the `aset_egen`-command. I am not sure how to determine a sensible number of paramters to be used in Cruickshank's equation 26, so at the moment, number of parameters is set to 0 causing Cruickshank eq. 27 (the one based on Rfree) to be used.
- fixed mistake in URL given in program startup message
- fixed bug concerning y-ticks.

Version β 0.1c, 1-Nov-2000

- The manual was updated
- If the keyword `check` in the `ddm`-command is set to `auto`, the program will try to find a consistent set of atoms for the subsequent calculations all by itself. This works in many cases, but not always. If it does not work, the selection can still be done by hand using `check=loose` or `check=strict`. I will try to make the algorithm more robust in the future, so please send me examples were the automatic selection does not work.
- Several *atom sets* can now be simultaneously selected by the `setl` keyword of various commands. The most important application of this is for difference distance matrices: if you select a list of *atom set via* the `setl` keyword, all pairwise difference distances will be printed. Each matrix will be put onto a separate sheet, producing a sort of book. See example on page 16.
- Atoms can now be selected based on their `element` or number in their *atom set*. This is very handy if you want to exclude hydrogens, see example on page, e.g. the following script will get rid of all hydrogen atoms found in a `pdb`-file:

```
aset_read(ifile="test.pdb");
aset_sele(sele = (element <> "H"));
aset_write(ofile="test_without_hydrogens.pdb");
```

- A much easier mechanism for selecting stretches of residues has been implemented, e.g. selecting residues 19 to 25, 36 to 42, 45 and 56 to 58 can now be done in a statement of the form: `sele = (resi in {19-25:36-42:45:56-58})`.
- Some facilities to work with multiple models from NMR spectroscopy have been included. The information about rmsd's from the NMR ensemble is translated into positional esd's using the `aset_egen`-command. Two models are available: `esd_model=rmsd` simply uses the rmsd as the coordinate uncertainty, and `esd_model=rmsd2` uses the square of the rmsd as the coordinate uncertainty. In both cases, the uncertainty is multiplied by the number given *via* the keyword `esd_fac`, to allow to put the coordinate error on some pseudo-absolute scale. This facility is useful, if you want to compare NMR-structures and X-ray structures. See under the `aset_egen`-command in the reference manual.
- A schematic representation of secondary structure elements can now be included into difference distance matrix plots. Helices are shown as open, sheets as filled boxes. Secondary structure can be defined using the `aset_amod`-command (see reference manual and section 2.3.1). If `HELIX` and `SHEET` records are found in a pdb-file, the information is used. The keyword `ss_plot` of the `ddm`-command is used to trigger plotting of the secondary structure. For example, see Figure 3.4.
- A number of small bugs have been fixed

Version β 0.1, 11-Jun-2000

- This is the very first beta-test release of ESCET. Please be patient and report all errors, problems, glitches, misunderstanding etc. p.p. to `trs(at)shelx.uni-ac.gwdg.de`.

Bibliography

[1] K. S. Anderson, E. W. Miles, and K. A. Johnson. Serine Modulates Substrate Channeling in Tryptophan Synthase. *J.Biol.Chem*, 266:8020–8033, 1991.

[2] D. W. J. Cruickshank. Remarks about protein structure precision. *Acta Cryst.*, D55:583–601, 1999.

[3] C. C. Hyde and E. W. Miles. The Tryptophan Synthase Multienzym Complex: Exploring the Structure-Function Relationships with X-ray Crystallography and Mutagenesis. *Bio/Technology*, 8:27–32, 1990.

[4] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Cryst.*, A32:922–923, 1976.

[5] W. Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Cryst.*, A34:827–828, 1978.

[6] T. R. Schneider. Objective comparison of protein structures: error-scaled difference distance matrices. *Acta Cryst.*, D56:714–721, 2000.

[7] T. R. Schneider. A genetic algorithm for the identification of conformationally invariant regions in protein molecules. *Acta Cryst.*, D58:195–208, 2002.

[8] T. R. Schneider. Domain identification by iterative analysis of error-scaled difference distance matrices. *Acta Cryst.*, D60:2269–2275, 2004.

[9] T. R. Schneider, E. Gerhardt, M. Lee, P. Lian, K. S. Anderson, and Schlichting I. Loop closure and intersubunit communication in tryptophan synthase. *Biochem.*, 37:5394–5406, 1998.